Fuzzy Entropy-Assisted Fuzzy-Rough Feature Selection

MacParthaláin, Neil Seosamh; Jensen, Richard; Shen, Qiang

DOI:
10.1109/FUZZY.2006.1681746

Publication date:
2006

Citation for published version (APA):
MacParthaláin, N. S., Jensen, R., & Shen, Q. (2006). *Fuzzy Entropy-Assisted Fuzzy-Rough Feature Selection*. Paper presented at Fuzzy Systems, Vancouver, Canada. https://doi.org/10.1109/FUZZY.2006.1681746

2006 IEEE International Conference on Fuzzy Systems
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

Fuzzy Entropy-Assisted Fuzzy-Rough Feature Selection

Neil Mac Parthaláin, Richard Jensen and Qiang Shen

Abstract— **Feature Selection (FS) is a dimensionality reduction technique that aims to select a subset of the original features of a dataset which offer the most useful information. The benefits of feature selection include improved data visualisation, transparency, reduction in training and utilisation times and improved prediction performance. Methods based on fuzzy-rough set theory (FRFS) have employed the dependency function to guide the process with much success. This paper presents a novel fuzzy-rough FS technique which is guided by fuzzy entropy. The use of this measure in fuzzy-rough feature selection can result in smaller subset sizes than those obtained through FRFS alone, with little loss or even an increase in overall classification accuracy.**

I. Introduction

The task of feature selection is to select a subset of the original features present in a given dataset which provides most of the useful information. Hence, after selection has taken place, most of the important information of the dataset should still remain. In fact, good FS techniques should be able to detect and ignore noisy and misleading features. The result of this, is that dataset quality may even *increase* after selection.

There are several potential benefits of feature selection:

1) *Facilitating data visualisation*. By reduction of the data to fewer dimensions, trends within the data can be more easily identified. This can be very important where only a few features have an influence on data outcomes.
2) *Reduction of measurement and storage requirements*. In domains where features correspond to particular measurements (for instance, a water treatment plant [15]), fewer features are highly desirable due to the expense and time-cost of taking such measurements.
3) *Reduction of training and utilisation times*. With smaller datasets, the runtimes of learning algorithms can improve significantly, for both training and classification phases.
4) *Improvements in prediction performance*. Classifier accuracy can be increased as a result of feature selection, through the removal of noisy or misleading features.

For those methods that extract knowledge from data (e.g. rule induction) the benefits of FS also include improvements in the readability of the discovered knowledge. When induction algorithms are applied to reduced data, the resulting rules are more compact. A good feature selection step will remove unnecessary attributes which may affect both rule comprehension and rule prediction performance.

The work on rough set theory (RST) offers an alternative, and formal methodology that can be employed to reduce dimensionality of datasets, as a preprocessing step to assist any chosen modelling method for learning from data. It helps to select the most information-rich features in a dataset, without transforming the data, whilst at the same time attempting to minimise information loss during the selection process. Computationally, the approach is highly efficient, relying on simple set operations, which makes it suitable as a preprocessor for techniques that are much more complex. Unlike statistical correlation-reduction approaches [5], RST requires no human input or intervention. Most importantly however, it retains the underlying semantics of the data, which results in models that are more transparent to human scrutiny.

It is most often the case that the values of attributes may be both crisp and *real-valued*, and this is where many feature selectors, particularly those based on traditional rough set theory, encounter a problem. It is not possible to determine whether two attribute values are similar and how far this similarity extends; for example, two close values may only differ as a result of noise, but in RST they are considered to be as different as two values of a different order of magnitude. One answer to this problem has been to discretise the dataset beforehand, thus producing a new dataset with crisp values. This however, is often still inadequate, as the degrees of membership of values to discretised values are not considered whatsoever. This consequently leads to information loss, which contradicts the rough set ideology of information content retention.

A solution to this problem is to use a fuzzy-rough approach. As discussed previously, RST can only operate effectively on datasets which contain discrete values, and there is no internal process which can be used to deal with *real-valued* and/or noisy data. Fuzzy-rough feature selection (FRFS) builds on rough set FS, but includes a fuzzification process which is carried out on the data of a given dataset. This fuzzification can be derived from the data itself, requiring no further information. This avoids any need for a separate data discretisation step of the rough set approach mentioned above, and hence associated information loss. Object memberships to the resulting fuzzy sets are used to guide the selection process, unlike the crisp method.

From previous experimentation with crisp rough sets and entropy [8] it was observed that entropy-based methods often

Neil Mac Parthaláin (email: nsm03@aber.ac.uk), Richard Jensen (email: rkj@aber.ac.uk), and Qiang Shen (email: qqs@aber.ac.uk), are with the Department of Computer Science, University of Wales, Aberystwyth, Wales, UK.

0-7803-9489-5/06/$20.00/©2006 IEEE

found smaller reducts than those based on the dependency function. This motivates a new fuzzy-rough technique using fuzzy entropy [10] to guide search, in order to locate optimal fuzzy-rough subsets.

The remainder of this paper is structured as follows. Section 2 summarises the theoretical basis and ideas of FRFS, along with a look at the fuzzy-rough QUICKREDUCT algorithm. Section 3 describes a fuzzy entropy-assisted approach to FRFS and corresponding algorithm. Section 4 shows the results of applying both FRFS, and entropy-based FRFS approaches to a number of datasets, along with a comparison of run times, classification accuracies, and dimensionality reduction. Section 5 concludes the paper along with suggestions for further work.

II. FUZZY-ROUGH FEATURE SELECTION

The principal focus of this paper lies in Fuzzy-entropy assisted FRFS (FEFRFS), however an in-depth view of the current FRFS methodology is necessary to appreciate the FEFRFS approach fully.

The rough set selection process described in [2] can only operate effectively with datasets containing discrete values. However, most datasets contain real-valued features and so it is necessary to perform a discretisation step beforehand. This is typically implemented by standard fuzzification techniques. As membership degrees of feature values to fuzzy sets are not exploited in the process of dimensionality reduction, important information has been lost. By employing *fuzzy-rough* sets, it is possible to use this information to better guide feature selection.

A fuzzy-rough set is defined by two fuzzy sets, fuzzy lower and upper approximations, obtained by extending the corresponding crisp rough set notions. In the crisp case, elements that belong to the lower approximation (i.e. have a membership of 1) are said to belong to the approximated set with absolute certainty. In the fuzzy-rough case, elements may have a membership in the range [0,1], allowing greater flexibility in handling uncertainty.

Fuzzy-Rough Feature Selection [8] is concerned with the reduction of information or decision systems through the use of fuzzy-rough sets. Let $I = (\mathbb{U}, \mathbb{A})$ be an information system, where $\mathbb{U}$ is a non-empty set of finite objects (the universe) and $\mathbb{A}$ is a non-empty finite set of attributes such that $a : \mathbb{U} \rightarrow V_a$ for every $a \in \mathbb{A}$. V_a is the set of values that attribute a may take. For decision systems, $\mathbb{A} = \{\mathbb{C} \cup \mathbb{D}\}$ where $\mathbb{C}$ is the set of input features and $\mathbb{D}$ is the set of decision values.

A. Fuzzy Equivalence Classes

Fuzzy equivalence classes [6], [12] are central to the fuzzy-rough set approach in the same way that crisp equivalence classes are central to classical rough sets. For typical applications, this means that the decision values and the conditional values may all be fuzzy. The concept of crisp equivalence classes can be extended by the inclusion of a fuzzy similarity relation S on the universe, which determines the extent to which two elements are similar in S. The usual properties of reflexivity ($\mu_S(x, x) = 1$), symmetry ($\mu_S(x, y) = \mu_S(y, x)$) and transitivity ($\mu_S(x, z) \geq \mu_S(x, y) \wedge \mu_S(y, z)$) hold.

Using the fuzzy similarity relation, the fuzzy equivalence class $[x]_S$ for objects close to x can be defined:

$$\mu_{[x]_S}(y) = \mu_S(x, y) \tag{1}$$

The following axioms should hold for a fuzzy equivalence class F:

- $\exists x$, $\mu_F(x) = 1$
- $\mu_F(x) \wedge \mu_S(x, y) \leq \mu_F(y)$
- $\mu_F(x) \wedge \mu_F(y) \leq \mu_S(x, y)$

The first axiom corresponds to the requirement that an equivalence class is non-empty. The second axiom states that elements in y's neighbourhood are in the equivalence class of y. The final axiom states that any two elements in F are related via the fuzzy similarity relation S. Obviously, this definition degenerates to the normal definition of equivalence classes when S is non-fuzzy. The family of normal fuzzy sets produced by a fuzzy partitioning of the universe of discourse can play the role of fuzzy equivalence classes [6].

B. Fuzzy Lower and Upper Approximations

The fuzzy lower and upper approximations are fuzzy extensions of their crisp counterparts. Informally, in crisp rough set theory, the lower approximation of a set contains those objects that belong to it with certainty. The upper approximation of a set contains the objects that possibly belong. From the literature, the fuzzy P-lower and P-upper approximations are defined as [6]:

$$\mu_{\underline{P}X}(F_i) = inf_x max\{1 - \mu_{F_i}(x), \mu_X(x)\} \quad \forall i \tag{2}$$

$$\mu_{\overline{P}X}(F_i) = sup_x min\{\mu_{F_i}(x), \mu_X(x)\} \quad \forall i \tag{3}$$

where $\mathbb{U}/P$ stands for the partition of the universe of discourse, $\mathbb{U}$, with respect to a given subset P of features, and F_i denotes a fuzzy equivalence class belonging to $\mathbb{U}/P$. Note that although the universe of discourse in feature reduction is finite, this is not the case in general, hence the use of sup and inf above. These definitions diverge a little from the crisp upper and lower approximations, as the memberships of individual objects to the approximations are not explicitly available. As a result of this, the fuzzy lower and upper approximations are redefined as [7]:

$$\mu_{\underline{P}X}(x) = \sup_{F \in \mathbb{U}/P} min(\mu_F(x), \inf_{y \in \mathbb{U}} max\{1 - \mu_F(y), \mu_X(y)\}) \tag{4}$$

$$\mu_{\overline{P}X}(x) = \sup_{F \in \mathbb{U}/P} min(\mu_F(x), \sup_{y \in \mathbb{U}} min\{\mu_F(y), \mu_X(y)\}) \tag{5}$$

The tuple $< \underline{P}X, \overline{P}X >$ is called a fuzzy-rough set. For this particular feature selection method, the upper approximation is not used, though this may be useful for other methods.

For an individual feature, a, the partition of the universe by $\{a\}$ (denoted $\mathbb{U}/IND(\{a\})$) is considered to be the set of those fuzzy equivalence classes for that feature. For example, if the two fuzzy sets N_a and Z_a are generated for feature a during fuzzification, the partition $\mathbb{U}/IND(\{a\})$ = $\{N_a, Z_a\}$. If the fuzzy-rough feature selection process is to be useful, it must be able to deal with multiple features, finding the dependency between various subsets of the original feature set. For instance, it may be necessary to be able to determine the degree of dependency of the decision feature(s) with respect to feature set P = $\{a, b\}$. In the crisp case, $\mathbb{U}/P$ contains sets of objects grouped together that are indiscernible according to both features a and b. In the fuzzy case, objects may belong to many equivalence classes, so the cartesian product of $\mathbb{U}/IND(\{a\})$ and $\mathbb{U}/IND(\{b\})$ must be considered in determining $\mathbb{U}/P$. In general,

$$\mathbb{U}/P = \otimes\{a \in P : \mathbb{U}/IND(\{a\})\} \tag{6}$$

For example, if P = $\{a, b\}$, $\mathbb{U}/IND(\{a\})$ = $\{N_a, Z_a\}$ and $\mathbb{U}/IND(\{b\})$ = $\{N_b, Z_b\}$, then

$$\mathbb{U}/P = \{N_a \cap N_b, N_a \cap Z_b, Z_a \cap N_b, Z_a \cap Z_b\}$$

Clearly, each set in $\mathbb{U}/P$ denotes an equivalence class. The extent to which an object belongs to such an equivalence class is therefore calculated by using the conjunction of constituent fuzzy equivalence classes, say F_i, $i = 1, 2, ..., n$:

$$\mu_{F_1 \cap ... \cap F_n}(x) = min(\mu_{F_1}(x), \mu_{F_2}(x), ..., \mu_{F_n}(x)) \tag{7}$$

C. Fuzzy-Rough Reduction Method

Fuzzy-Rough Feature Selection builds on the notion of the fuzzy lower approximation to enable reduction of datasets containing real-valued features. The process becomes identical to the crisp approach when dealing with nominal well-defined features.

The crisp positive region in the standard RST is defined as the union of the lower approximations. By the extension principle, the membership of an object $x \in \mathbb{U}$, belonging to the fuzzy positive region can be defined by

$$\mu_{POS_P(Q)}(x) = \sup_{X \in \mathbb{U}/Q} \mu_{\underline{P}X}(x) \tag{8}$$

Object x will not belong to the positive region only if the equivalence class it belongs to is not a constituent of the positive region. This is equivalent to the crisp version where objects belong to the positive region only if their underlying equivalence class does so.

Using the definition of the fuzzy positive region, a new dependency function between a set of features Q and another set P can be defined as follows:

$$\gamma'_P(Q) = \frac{|\mu_{POS_P(Q)}(x)|}{|\mathbb{U}|} = \frac{\sum_{x \in \mathbb{U}} \mu_{POS_P(Q)}(x)}{|\mathbb{U}|} \tag{9}$$

As with crisp rough sets, the dependency of Q on P is the proportion of objects that are discernible out of the entire dataset. In the present approach, this corresponds to determining the fuzzy cardinality of $\mu_{POS_P(Q)}(x)$ divided by the total number of objects in the universe.

FRQUICKREDUCT($\mathbb{C}$,$\mathbb{D}$).
$\mathbb{C}$, the set of all conditional features;
$\mathbb{D}$, the set of decision features.

(1) $R \leftarrow \{\}$, $\gamma'_{best} \leftarrow 0$, $\gamma'_{prev} \leftarrow 0$
(2) **do**
(3) $T \leftarrow R$
(4) $\gamma'_{prev} \leftarrow \gamma'_{best}$
(5) $\forall x \in (\mathbb{C} - R)$
(6) **if** $\gamma'_{R \cup \{x\}}(\mathbb{D}) > \gamma'_T(\mathbb{D})$
(7) $T \leftarrow R \cup \{x\}$
(8) $\gamma'_{best} \leftarrow \gamma'_T(\mathbb{D})$
(9) $R \leftarrow T$
(10) **until** $\gamma'_{best} == \gamma'_{prev}$
(11) **return** R

Fig. 1. The fuzzy-rough QUICKREDUCT algorithm

A fuzzy-rough QUICKREDUCT algorithm, based on the crisp version [2], has been developed as given in Fig. 1. It employs the fuzzy-rough dependency function γ' to choose which features to add to the current reduct candidate. The algorithm terminates when the addition of any remaining feature does not increase the dependency. As with the original algorithm, for a dimensionality of n, the worst case dataset will result in $(n^2 + n)/2$ evaluations of the dependency function. However, as fuzzy-rough set-based feature selection is used for dimensionality reduction prior to any involvement of the system which will employ those features belonging to the resultant reduct, this operation has no negative impact upon the run-time efficiency of the system.

III. FUZZY ENTROPY-ASSISTED FRFS

Fuzzy Entropy-assisted FRFS uses the FRFS methodology as a basis for dimensionality reduction, while using a fuzzy-entropy measure to guide the FS process, rather than the dependency function value as described in the previous section.

A. Classical and Information Entropy (IE)

Classical Entropy may be defined as a measure of the degradation or dispersal of energy and also as the energy form of a system that relates to its internal state of disorder or randomness. Entropy may also be described as a measure of progress of a process of equalisation. It is often used in relation to thermodynamic or metabolic biological processes. High entropy values are indicative of disordered states, and low entropy values are characteristic of ordered states.

Information entropy (IE) or Shannon entropy [14] is also a measure of the amount of disorder in a system and can be defined as:

$$H(X) = -\sum_{i=0}^{N} p_i \log_2 p_i \tag{10}$$

The entropy of the event *X* is the sum, over all possible outcomes *i* of *X* , of the product of the probability of outcome *i* times the log of the probability of *i*. This can also be applied to a general probability distribution, rather than a discrete-valued event.

The IE value tends to zero with increasing order in any system. It is interesting to note at this point that the fuzzy-rough dependency function value tends to 1 with any increase in order. Having considered this fact, the motivation for investigation of a fuzzy entropy-based approach may not be clear. However, as noted previously, the use of fuzzy-entropy-based techniques often discovered smaller reducts than dependency function-based methods [8].

A fuzzy entropy-assisted approach selects subsets with respect to their entropy value and uses this value to guide the feature selection process.

B. Fuzzy Entropy Measure

Again, let $I = (\mathbb{U}, \mathbb{A})$ be a decision system, where $\mathbb{U}$ is a non-empty set of finite objects. $\mathbb{A} = \{\mathbb{C} \cup \mathbb{D}\}$ is a non-empty finite set of attributes, where $\mathbb{C}$ is the set of input features and $\mathbb{D}$ is the set of classes. An attribute $a \in \mathbb{A}$ has corresponding fuzzy subsets $F_1, F_2, ..., F_n$. The fuzzy entropy for a fuzzy subset F_i can be defined as:

$$H(F_i) = -\sum_{D \in \mathbb{U}/\mathbb{D}} p(D|F_i) \log_2 p(D|F_i) \tag{11}$$

where, $p(D|F_i)$ is the relative frequency of the fuzzy subset F_i of attribute a with respect to the decision D, and is defined:

$$p(D|F_i) = \frac{|D \cap F_i|}{|F_i|} \tag{12}$$

The cardinality of a fuzzy set is denoted by $|\cdot|$. Based on these definitions, the fuzzy entropy for an attribute subset R is defined as follows:

$$E(R) = \sum_{F_i \in \mathbb{U}/R} \frac{|F_i|}{\sum_{Y_i \in \mathbb{U}/R} |Y_i|} H(F_i) \tag{13}$$

This fuzzy entropy can be used to gauge the utility of attribute subsets in a similar way to that of the fuzzy-rough measure. However, the fuzzy entropy measure decreases with increasing subset utility, whereas the fuzzy-rough dependency measure increases. With these definitions, a new feature selection mechanism can be constructed that uses fuzzy entropy to guide the search for the best fuzzy-rough feature subset.

FREQUICKREDUCT($\mathbb{C}$,$\mathbb{D}$).
$\mathbb{C}$, the set of all conditional features;
$\mathbb{D}$, the set of decision features.

(1) $T \leftarrow \{\}$, $\gamma'_{prev} \leftarrow 0$
(2) **do**
(3) $R \leftarrow T$
(4) $\gamma'_{prev} \leftarrow \gamma'_T(\mathbb{D})$
(5) $\forall x \in (\mathbb{C} - R)$
(6) **if** $E(R \cup \{x\}) < E(T)$
(7) $T \leftarrow R \cup \{x\}$
(8) **until** $\gamma'_T(\mathbb{D}) \leq \gamma'_{prev}$
(9) **return** R

Fig. 2. The fuzzy-rough fuzzy entropy-based QUICKREDUCT algorithm

C. Fuzzy-Rough Entropy-based QUICKREDUCT

Figure 2 below shows a fuzzy-rough entropy-based QUICKREDUCT algorithm based on the previously described fuzzy-rough algorithm in figure 1.

FREQUICKREDUCT is similar to the fuzzy-rough algorithm but uses the entropy value of a data subset to guide the feature selection process. If the fuzzy entropy value of the current reduct candidate is smaller than the previous, then this reduct is retained and used in the next iteration of the loop. It is important to point out that the reduct is evaluated by examining its entropy value, termination only occurs when the addition of any remaining features results in a decrease in the dependency function value (γ'_{prev}). The fuzzy-entropy value therefore is not used as a termination criteria.

The algorithm begins with an empty subset R and with γ'_{prev} initialised to zero. The do-until loop works by examining the entropy value of a subset and incrementally adding one conditional feature at a time, until the dependency function value begins to fall to a value that is lower or equal to that of the last subset. For each iteration, a conditional feature that has not already been evaluated will be temporarily added to the subset R. The entropy of the subset currently being examined (5) is then evaluated and compared with the entropy of T, (the previous subset). If the entropy value of the current subset is lower (6), then the attribute added in (5) is retained as part of the new reduct T (7).

The loop continues to evaluate in the above manner by adding conditional features, until the dependency value of the current reduct candidate ($\gamma'_R(\mathbb{D})$) falls to a value lower than or equal to that of the previously evaluated reduct candidate.

D. A Worked Example

To illustrate the operation of the new fuzzy entropy-based algorithm, a small example dataset (given in table I) is considered, containing real-valued conditional attributes with nominal decisions.

Table I contains three real-valued conditional attributes and a crisp-valued decision attribute. To begin with, the algorithm initializes the potential reduct (i.e. the current best set of attributes) to the empty set.

Object	a	b	c	q
1	−0.4	−0.3	−0.1	no
2	−0.4	0.2	−0.2	yes
3	−0.3	−0.4	−0.1	no
4	0.3	−0.3	0	yes
5	0.2	−0.3	0	yes
6	0.2	0	0	no

TABLE I
EXAMPLE DATASET: CRISP DECISIONS

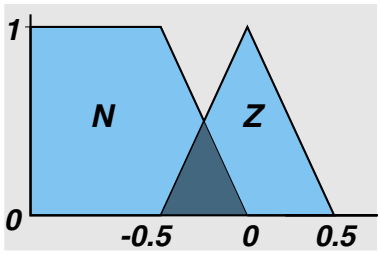

Fig. 3. Fuzzifications for conditional features

Using the fuzzy sets defined in figure 3 (for all conditional attributes), and setting $A = \{a\}$, $B = \{b\}$, $C = \{c\}$ and $\mathbb{D} = \{q\}$, the following equivalence classes are obtained:

$$\begin{array}{lcl} \mathbb{U}/A & = & \{N_a, Z_a\} \\ \mathbb{U}/B & = & \{N_b, Z_b\} \\ \mathbb{U}/C & = & \{N_c, Z_c\} \\ \mathbb{U}/\mathbb{D} & = & \{\{1,3,6\},\{2,4,5\}\} = \{D_1, D_2\} \end{array}$$

The algorithm begins with an empty subset, and considers the addition of individual features. The attribute that results in the greatest decrease in fuzzy entropy will ultimately be added to the reduct candidate. For attribute a, the fuzzy entropy is calculated as follows ($A = \{a\}$):

$$E(A) = \frac{|N_a|}{|N_a| + |Z_a|} H(N_a) + \frac{|Z_a|}{|N_a| + |Z_a|} H(Z_a)$$

For the first part of the summation, the value $H(N_a)$ must be determined. This is achieved in the following way:

$$\begin{array}{lcl} H(N_a) & = & -\sum_{D \in \mathbb{U}/\mathbb{D}} p(D|N_a) \log_2 p(D|N_a) \\ & = & -p(D_1|N_a) \log_2 p(D_1|N_a) \\ & & + -p(D_2|N_a) \log_2 p(D_2|N_a) \end{array}$$

The required probabilities are $p(D_1|N_a)$ = 0.6363637, $p(D_2|N_a)$ = 0.3636363. Hence, $H(N_a)$ = 0.94566023. In a similar way, $H(Z_a)$ can be calculated, giving a value of 1.0.

To determine the fuzzy entropy for a, the values $\frac{|N_a|}{|N_a|+|Z_a|}$ and $\frac{|Z_a|}{|N_a|+|Z_a|}$ must also be determined. This is achieved through the standard fuzzy cardinality, resulting in a fuzzy entropy value of:

$$\begin{array}{lcl} E(A) & = & (0.47826084.H(N_a)) + (0.5217391.H(Z_a)) \\ & = & (0.47826084 \times 0.94566023) \\ & & + (0.5217391 \times 1.0) \\ & = & 0.9740114 \end{array}$$

Repeating this process for the remaining attributes gives:

$$\begin{array}{lcl} E(B) & = & 0.99629750 \\ E(C) & = & 0.99999994 \end{array}$$

From this it can be seen that attribute a will cause the greatest decrease in fuzzy entropy. This attribute is chosen and added to the potential reduct, $R \leftarrow R \cup \{a\}$. This subset is then evaluated using the fuzzy-rough dependency measure, resulting in $\gamma_R(\mathbb{D}) = 0.3333333$. The previous dependency value is 0 (the algorithm started with the empty set), hence the search continues. The process iterates and the two fuzzy entropy values calculated are

$$\begin{array}{lcl} E(\{a,b\}) & = & 0.7878490 \\ E(\{a,c\}) & = & 0.9506136 \end{array}$$

Adding attribute b to the reduct candidate causes the larger decrease of fuzzy entropy, so the new candidate becomes $\{a,b\}$. The resulting dependency value for this, $\gamma_{\{a,b\}}(\mathbb{D})$, is 0.56666666. This is, again, larger than the previous dependency value, and so search continues. Lastly, attribute c is added to the potential reduct:

$$\begin{array}{lcl} E(\{a,b,c\}) & = & 0.7412282 \\ (\gamma_{\{a,b,c\}}(\mathbb{D}) & = & 0.56666666) \end{array}$$

As this causes no increase in dependency, the algorithm stops and outputs the reduct $\{a,b\}$. The dataset can now be reduced to only those attributes appearing in the reduct.

IV. EXPERIMENTATION

This section presents the results of experimental studies using the datasets described in table II. These datasets are small-to-medium in size, with between 120 and 390 objects per dataset and feature sets ranging from 5 to 39. All datasets have been obtained from [1] and [11]. A comparison of the entropy and FRFS-based dimensionality reduction techniques is given based on classification accuracy, reduct size and time taken.

The method employed uses a pre-categorisation step which generates associated fuzzy sets for a dataset. The FS process then generates a reduced dataset and associated reduced fuzzy sets. These reduced datasets are then classified using the relevant classifier (J48, JRip, PART). (Note that the FS step is not employed for the unreduced dataset)

A. Classifiers

In the generation of results for classification accuracies, three classifiers were employed – J48, JRip, and PART [17].

J48 [13] creates decision trees by choosing the most informative features and recursively partitioning the data into subtables based on their values. Each node in the tree represents a feature with branches from a node representing the alternative values this feature can take according to the current subtable. Partitioning stops when all data items in the subtable have the same classification. A leaf node is then created, and this classification assigned.

Dataset	Objects	Features	Decision feat. type	Description
water 2	390	39	binary	Water treatment database
water 3	390	39	3-class	Water treatment database
cleveland	297	14	binary	Heart Disease database
glass	214	10	6-class	Glass identification database
heart	270	14	binary	Heart Disease database
ionosphere	230	35	binary	Ionosphere radar classification
iris	150	5	3-class	Plant classification database
olitos	120	26	4-class	Chemical analysis database
wine	178	14	3-class	Wine recognition database

TABLE II
DATASET DESCRIPTION

	J48			JRip			PART		
Dataset	Unreduced	FRFS	Entropy	Unreduced	FRFS	Entropy	Unreduced	FRFS	Entropy
water 2	83.33	80.26	81.79	83.85	84.36	84.36	85.64	82.56	85.38
water 3	77.44	79.74	78.46	81.28	82.05	84.62	79.49	78.97	81.28
cleveland	51.85	55.22	52.53	52.19	54.55	53.87	50.17	52.19	51.52
glass	67.29	69.63	69.63	71.50	69.63	69.63	67.76	68.22	68.22
heart	76.67	78.89	78.52	77.41	78.89	82.59	73.33	78.52	80.0
ionosphere	87.83	91.30	88.7	86.52	87.83	89.13	88.27	91.30	89.57
iris	96.00	96.00	96.00	94.00	94.00	95.33	94.00	94.00	95.33
olitos	67.50	67.50	68.33	70.83	70.83	67.50	57.50	62.50	70.83
wine	94.38	92.14	93.26	92.70	88.76	89.89	93.82	93.82	91.57

TABLE III
AVERAGE CLASSIFICATION ACCURACY

	Original number	Reduct size		Final dependency value		Time taken to locate reduct		Time to build model	
Dataset	of features	FRFS	Entropy	FRFS	Entropy	FRFS	Entropy	FRFS	Entropy
water 2	39	11	8	0.588	0.540	96.58	68.29	0.034	0.027
water 3	39	12	11	0.595	0.549	158.73	1657.44	0.08	0.067
cleveland	14	11	10	0.516	0.535	24.11	130.63	0.09	0.1
glass	10	9	9	0.359	0.359	1.61	4.89	0.047	0.13
heart	14	11	9	0.578	0.607	11.84	56.48	0.027	0.074
ionosphere	35	11	11	0.673	0.677	61.80	962.56	0.087	0.037
iris	5	5	3	0.707	0.658	0.031	0.031	0.001	0.007
olitos	26	10	8	0.572	0.620	11.20	22.36	0.023	0.02
wine	14	10	9	0.844	0.862	1.42	9.83	0.023	0.013

TABLE IV
COMPARISON OF REDUCT SIZE, DEPENDENCY VALUE, & RUN TIMES

JRip [3] learns propositional rules by repeatedly growing rules and pruning them. During the growth phase, antecedents are added greedily until a termination condition is satisfied. Antecedents are then pruned in the next phase subject to a pruning metric. Once the ruleset is generated, a further optimization is performed where rules are evaluated and deleted based on their performance on randomized data.

PART [18] generates rules by means of repeatedly creating partial decision trees from data. The algorithm adopts a divide-and-conquer strategy such that it removes instances covered by the current ruleset during processing. Essentially, a rule is created by building a pruned tree for the current set of instances; the leaf with the highest coverage is promoted to a rule.

B. Comparison of Classification Accuracy

The data which are presented in table III shows the average classification accuracy as a percentage obtained using the 10-fold cross validation method. The classification was initially performed on the unreduced dataset, followed by the reduced datasets which were obtained by using both FRFS, and entropy-based dimensionality reduction techniques.

In most cases the classification accuracy increases or remains at the same level (for both FRFS and entropy-based methods). There are some notable exceptions however, where a decrease in classification accuracy is observed. When such decreases are compared to the reduction in dimensionality achieved using entropy-based reduction techniques, it is apparent that they are not significant. For example, for the data set water 2 in table IV, the reduction of the feature set from 39 to 11 (for both FRFS and entropy-based techniques)

translates to an overall reduction in dimensionality of 71.8%. The corresponding decrease in classification accuracy however is only in the order of 1.9%.

When comparing both entropy and FRFS-based techniques, the results show that for the entropy-based approach, there are more instances of increases in classification than for the same datasets using the FRFS based approach.

In summary, although there are some instances where the classification accuracy may decrease slightly, the general trend is to an increase in classification accuracy. Where there is a decrease, it is small in comparison to the overall reduction of dimensionality.

The results for JRip show that it has fewer instances of increase in classification accuracy than either J48 or PART. Indeed there are instances where the accuracy has decreased when applied to the reduced datasets for both FRFS and entropy-based methods.

The J48 classifier offers improved results over JRip but still has some instances where the classification values decrease when classifying the reduced datasets.

The classifier results for PART show the most consistent increase in classification accuracy which shows improvement on the JRip and J48 classifiers.

C. Reduct Size, Runtimes, and Dependency Function

Presented in table IV is a comparison of reduct size, dependency value and runtime data, using both FRFS and entropy-based approaches.

There is an obvious and clear advantage to the entropy-based approach in relation to reduct size. The entropy-based method consistently returns reducts that are at least equal in size to the reduct returned by the FRFS method, but usually smaller. In fact the entropy-based approach returns reducts that are smaller in size for nearly 78% of the datasets listed, – there are only 2 cases where reducts are equal in size to those returned by the FRFS method.

It is clear from the data that the entropy-based technique runtimes are considerably longer than the FRFS-based method. The water 3 database is one particular example that demonstrates this - the FRFS method takes 158.73s to run while the entropy-based method takes 1567.44s (15.6 min) - nearly 91% longer. The computational overhead of the entropy-based method in comparison to FRFS is significant, and it must also be considered that no attempt has been made to optimise the FEFRFS algorithm.

As mentioned previously, the overall runtime efficiency of both approaches can be summarised by the times returned for the water 3 dataset, which is considerably larger than the FRFS-based approach. This indicates that on average the entropy-based method takes significantly longer to discover reducts than the FRFS-based approach.

Whilst the entropy-based FS approach is guided by the fuzzy entropy value, it would be expected that the dependency function value results would not reach the same level as those of the FRFS method (as the FRFS method is guided solely by the dependency function value). However, the average dependency function values returned for both FRFS and entropy-based approaches are almost indistinguishable (FRFS–0.603 and entropy-based–0.601), with some results for the entropy-based method even returning higher individual values.

When considering the relationship of the dependency function value to the classification accuracy. There is a trend towards an increase of dependency function value with increases in classification accuracy. Figures 4 and 5 show this clearly, however once again the entropy-based method shows a small but clear improvement over the FRFS method.

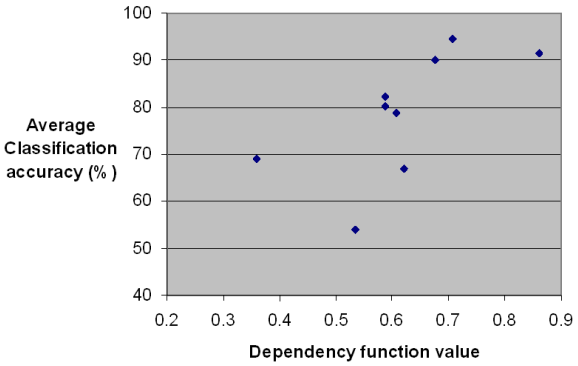

Fig. 4. FRFS Approach–Average Classification and Dependency Function value

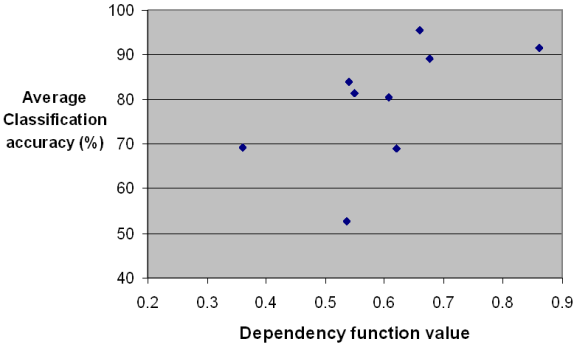

Fig. 5. Entropy-based Approach–Average Classification and Dependency Function value

V. Conclusions

Comparison of both FRFS and fuzzy entropy-based FRFS has shown that although the fuzzy entropy-based method will find smaller reducts than FRFS, this often occurs at the expense of runtime. However, it must be remembered that the entropy-based method is computationally more complex than FRFS.

Classification accuracy on average has been shown to increase or remain at the same level using the fuzzy entropy-based method. Where a decrease has been observed in relation to FRFS, it has been small and has always resulted in reducts that are smaller than the corresponding FRFS

reducts. As discussed previously, the actual decrease in value of classification accuracy is not significant.

The dependency function values of the entropy-based method are very close to those of the FRFS method but also marginally lower. When the average over all of the datasets is considered this is to be expected as the entropy-based approach is not guided using the dependency function.

It is clear from the results obtained in the previous section that an increase in the efficiency of the fuzzy entropy-based algorithm is highly desirable. The experimental work detailed in this paper did not take advantage of any optimisation for the fuzzifications or classifiers. It is expected that the results obtained through the use of optimisation would reflect a marked improvement. Future work would include the implementation of such optimisation methods for the fuzzy entropy-based algorithm. Also, due to the fuzzy nature of the data examined during experimentation, the use of fuzzy classifiers in place of the above listed classifiers would lead to an increase in classification accuracy.

Other areas which warrant further investigation include alternatives to the greedy forward selection approach employed here. Investigations have already been carried out into the use of an ant colony optimization-based search mechanism for FRFS [9]. This approach was shown to be superior for locating optimal reducts. It is likely that similar improvements can be obtained for FEFRFS when adopting this search strategy.

REFERENCES

[1] C. Armanino, R. Leardi, S. Lanteri, and, G. Modi Chemom. Intell. Lab. Syst., vol. 5, pp. 343–354. 1989.

[2] A. Chouchoulas and Q. Shen. Rough set-aided keyword reduction for text categorisation. Applied Artificial Intelligence, Vol. 15, No. 9, pp. 843–873. 2001.

[3] W.W. Cohen. Fast effective rule induction. In Machine Learning: Proceedings of the 12th International Conference, pp. 115–123. 1995.

[4] S. M. Chen and J. D. Shie, A new method for feature subset selection for handling classification problems, Proceedings of the 2005 IEEE International Conference on Fuzzy Systems, Reno, Nevada, US, pp. 183–188, May 2005.

[5] P. Devijver and J. Kittler. Pattern Recognition: A Statistical Approach. Prentice Hall. 1982.

[6] D. Dubois and H. Prade. Putting rough sets and fuzzy sets together. In [16], pp. 203–232. 1992.

[7] R. Jensen and Q. Shen. Fuzzy-Rough Attribute Reduction with Application to Web Categorization. Fuzzy Sets and Systems, Vol. 141, No. 3, pp. 469–485. 2004.

[8] R. Jensen and Q. Shen. Semantics-Preserving Dimensionality Reduction: Rough and Fuzzy-Rough Based Approaches. IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 12, pp. 1457–1471. 2004.

[9] R. Jensen and Q. Shen. Fuzzy-Rough Data Reduction with Ant Colony Optimization. Fuzzy Sets and Systems, 149(1):5–20. 2005.

[10] B. Kosko. Fuzzy entropy and conditioning. Information Sciences, Vol. 40, No. 2, pp. 165–174. 1986.

[11] D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz, UCI Repository of machine learning databases [http://www.ics.uci.edu/ mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science. 1998.

[12] S.K. Pal and A. Skowron (Eds.). Rough-Fuzzy Hybridization: A New Trend in Decision Making. Springer Verlag, Singapore. 1999.

[13] J.R. Quinlan. C4.5: Programs for Machine Learning. The Morgan Kaufmann Series in Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA. 1993.

[14] C.E. Shannon, A mathematical theory of communication. Bell System Technical Journal, vol. 27, pp. 379–423, and pp. 623-656, July and October, 1948.

[15] Q. Shen and R. Jensen. Selecting Informative Features with Fuzzy-Rough Sets and its Application for Complex Systems Monitoring. Pattern Recognition, Vol. 37, No. 7, pp. 1351–1363. 2004.

[16] R. Slowinski, editor. Intelligent Decision Support. Kluwer Academic Publishers, Dordrecht. 1992.

[17] I.H. Witten and E. Frank. Data Mining: Practical machine learning tools with Java implementations. Morgan Kaufmann, San Francisco. 2000.

[18] I.H. Witten and E. Frank. Generating Accurate Rule Sets Without Global Optimization. In Machine Learning: Proceedings of the 15th International Conference, Morgan Kaufmann Publishers, San Francisco. 1998.